Autonomous Vocal and Backing Track Mixing

kelian Li
Center for Music Technology
Georgia Institute of Technology
Atlanta, USA
kli421@gatech.edu

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
Atlanta, USA
alexander.lerch@gatech.edu

***Abstract*—The field of automatic mixing is growing over the years, and the approaches are moving from rule-based expert systems to deep learning. While most research focuses on automatic mixing on multi-tracks, the area of vocal and backing track mixing is largely ignored. A rule-based baseline implementation closes the gap by performing autonomous level balance, compression, equalization, and reverberation. A trial experiment of level balance by deep learning shows that the model is not yet learning.**

***Index Terms*—automatic mixing, intelligent music production**

I. Introduction

Music mixing is the process of blending multitrack recordings through manipulating audio effects, and the mixing quality is ultimately judged by the preference of listeners. Automatic mixing aims to reach a similar level of quality by computational systems without human interventions [1]. Automatic mixing is a fairly new research field that can possibly, or has already, changed the music production process from professional audio engineers to amateur music creators. The research can not only increase the efficiency, but also help understand mixing from the aspects of both science and arts which may lead to innovations in music production.

This work intends to tackle the problem of vocal and backing track mixing by deep learning. Previous research either attempted to create single unified general-purposed systems which has not yet succeed, or targeted on a more specific scenario like audio mastering. The proposed project focus on vocal and backing track mixing which has not been addressed in previous research but with applications in professional music production and public music participation.

Derived from hip-hop culture, beats essentially mean the pre-made accompaniment tracks [2]. Unlike the traditional song-writing and music production where these two processes go hand in hand, communication and revision between beat makers and the artists are not necessary. This new norm of music making has expanded out of hip-hop community and adapted by the general music industry as shown in the success of beat-selling platforms like BeatStars[1] [3]. Automatic mixing can accelerate the mixing process and reduce the listening fatigue.

Another potential application is karaoke. Invented back in 1960's, karaoke has become a global phenomenon and brought joy to people for decades. With the rise of smartphones, mobile karaoke apps enable the users to sing and share their performances on social media [4]. Most karaoke mobile apps have some basic vocal mixing features. But without much knowledge of music mixing, general users are unable to produce very good mixes.

[1]BeatStars, http://www.beatstars.com/

II. Related work

Early automatic mixing methods are mostly rule-based expect systems for one type of audio effects. They incorporate some common practices by human mixing engineers into the algorithms that can perform similarly. Reiss [5] firstly introduced the concept of adaptive audio effects, and the key idea is that the parameters of an audio effects are modified by analysis of input audio through the feature extraction and mapping. The features in use range from low-level features as RMS and crest factor to high-level features as downbeats. Most early expert systems are based on or extended from adaptive audio effects. These systems are typically designed for one audio effect and any unspecified music sources. One example of such system for stereo panning is described by Perez Gonzalez and Reiss [6]. One of the rules says that low-frequency content should be kept in the center, so a filter bank and a peak meter are implemented to determine whether an audio source is dominated by low-frequency content. More comprehensive overview of rule-based automatic mixing systems can be found in the publishing by De Man et al. [7], Moffat and Sandler [1], and De Man et al. [8].

As described in the recent book chapter by Moffat [9], the major limitation of the data-driven automatic mixing is the data collection of mixing parameters. One obvious solution is to let users create training data [10] [11]. Another common solution is to adapt the reverse engineering approach [12] [13] as applied in [14] for drum level extraction and balancing. This approach requires the raw multi-track recordings and the corresponding professional mix-downs. So far the reverse engineering approach only covers a few limited audio effects.

It is possible to skip the process of extracting the mixing parameters by end-to-end data transformation. Martínez Ramírez and Reiss [15] present a proof-of-concept research using a deep autoencoder. Later on, Martínez Ramírez et al. [16] demonstrate an end-to-end intelligent mixing system for drum tracks with Wave-U-Net [17]. This system learns the

full mixing process of drum mixing and the result is indistinguishable from the professional engineer-generated mix. One drawback is that the end-to-end framework prevents further parameter control by humans. Steinmetz et al. [18] incorporate differentiable digital signal processing [19] to extract the mixing parameter data, and thus their system can produce human-readable mixing parameters. However, the implementation is very challenging, and the listening test result is not ideal. Martínez Ramírez et al. [20] propose a method to extract the mixing parameters for arbitrary black-box audio effects by gradient approximation. This method does not need dedicated mix reverse engineering for specific audio effects nor differentiable digital signal processing implementation. The result is promising on two single-track tasks: non-speech sound removal and music mastering.

On the one hand, despite the significant progress of deep learning in intelligent mixing, the lack of audio data continues to be the barrier for creating a general-purposed automatic mixing system. One the other hand, the data-driven approach starts to show some convincing outcomes in specific tasks like drum mixing and single-track processing.

Outside of academia, companies have begun adapting automatic mixing into their products, and the list includes the multi-effect plugin Neutron[2] and Balancer[3], and an online DAW Faders[4]. In the task of automatic mastering, Landr[5] SoundCloud (powered by Dolby)[6], Ozone[7], BandLab[8], and eMastered[9] are considered successful for achieving the commercially usable quality. Among all the popular karaoke apps, ChangBa[2] provides the automatic mixing service for level balance, equalization, and reverberation. Other apps like Smule[3] may include automatic mixing features that are not visible to the users.

III. Method

Due to the scope of this paper and the creative nature of music mixing, the audio effects to be considered are only level balance, compression, equalization, and reverberation. For each, a rule-based system is implemented as the baseline.

A. Rule-based Baseline System

1) Level Balance: A popular assumption of mixing also indicates that the lead vocal as the main element should be louder than other elements [21]. De Man et al. [22] conducted a mixing experiment and gathered the audio features of processed tracks. They found that the relative loudness of the vocal track and the full mix loudness is about -3 dB, in the range of -2.7 ± 1.6 dB.

In the implementation, the integrated loudness [23] of the lead vocal and the backing track are measured, and a gain change applies to the vocal accordingly. The targeted related loudness of the vocal and the backing track is set to -0.5 dB, and the resulted relative loudness of the vocal track and the full mix falls near -3 dB.

2) Compression: The dynamic range is measured by EBU dynamic range [24]. The average loudness range of the mixed vocal tracks in the MUSDB18HQ dataset [25] is 15.7 dB.

An iterative process optimize the vocal loudness range to 15.7 dB ± 1 dB by modifying the compression threshold and compression ratio. Other mixing parameters are unchanged. The attack time and the release time are set to 30 ms and 200 ms. The vocal signal is normalized before the loudness range measurement. The compression threshold and the compression ratio are initialized to 0 dB and 3:1. For each iteration, the compression threshold is reduced by 3 dB, and the compression ratio is increased by 0.1. The process ends if the loudness range is below 16.7 dB, or the current compression threshold is less than -35 dB. The open-source compressor plugin of choice is OmniCompressor[10] by IEM.

Algorithm 1 frequency unmasking

apply noise gate to lead vocal vox and the backing track bac

VOX_{band}, BAC_{band} = frequencyBandRMS(vox, bac)
$DIFF_{band}$ = $VOX_{band} - BAC_{band}$
$DIFF_{band}$ $-=$ mean($DIFF_{band}$)

vox^*, acc^* = KFilter(vox, bac)
VOX^*_{band}, BAC^*_{band} = frequencyBandRMS(vox^*, bac^*)
VOX_{idx} = argsort(VOX^*_{band})
BAC_{idx} = argsort(BAC^*_{band})

if idx in top 4 VOX_{idx} **then**
  if idx not in top 3 BAC_{idx} **then**
    if $DIFF_{band}[idx] < 0$ **then**
      $gain = DIFF_{band}[idx]$
      $freq = band[idx]$
      apply peak filter on vox
    end if
  end if
end if

if idx in top 3 ACC_{idx} **then**
  if idx not in top 4 VOX_{idx} **then**
    if $DIFF_{band}[idx] > 0$ **then**
      $gain = DIFF_{band}[idx]$
      $freq = band[idx]$
      apply peak filter on vox
    end if
  end if
end if

[2]Neutron, https://www.izotope.com/en/products/neutron/features/mix-assistant.html
[3]Balancer, https://www.sonible.com/balancer/
[4]Faders, https://faders.io/
[5]Landr, https://www.landr.com/
[6]SoundCloud (powered by Dolby), https://community.soundcloud.com/mastering-on-soundcloud
[7]Ozone, https://www.izotope.com/en/products/ozone/features/master-assistant.html
[8]BandLab, https://www.bandlab.com/mastering
[9]eMastered, https://emastered.com/
[10]IEM OmniCompressorhttps://plugins.iem.at/docs/plugindescriptions/

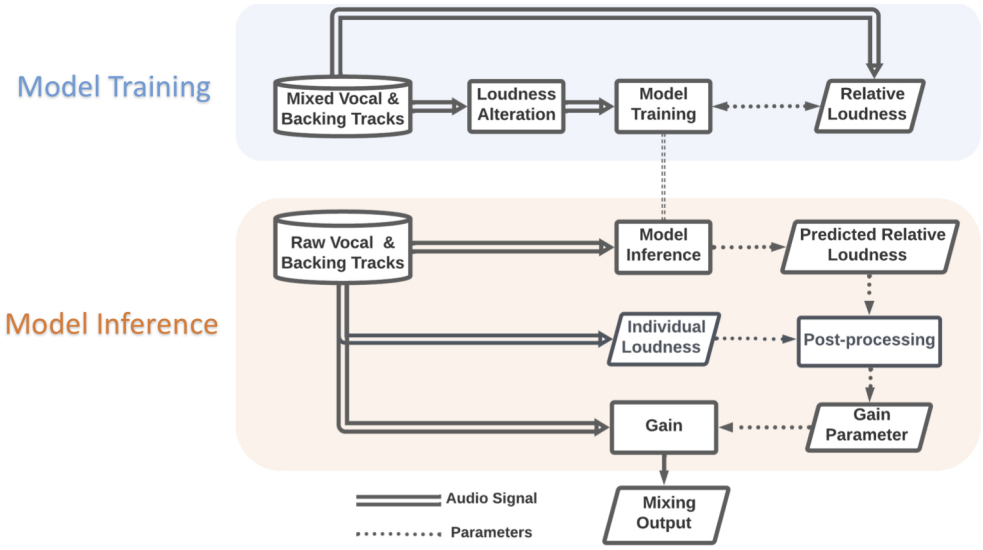

Fig. 1: Data-driven level balance approach

3) Equalization: The automatic equalization is achieved by the frequency masking reduction method proposed by Hafezi and Reiss [26]. Masking is defined as the process by which the audibility of one sound (the maskee) is reduced by the presence of another sound (the masker). Equalizers are commonly used to reduce the spectral masking in mixing.

There are three criteria to be met before a peak filter is applied to the masker signal at a given frequency band: (1) the frequency band is the essential band of the maskee signal; (2) the frequency band is the nonessential band of the masker signal; (3) the masker signal masks the maskee signal in this frequency band. In our case, the vocal and the backing track can be both the masker and the maskee at the same time, but the filter is only applied to the vocal. When the vocal is the masker, the peak filter applies a negative gain (cut). When the vocal is the maskee, the peak filter applies a positive gain (boost). The center frequency of 10 octave bands in use are [31.5, 63, 125, 250, 500, 1K, 2K, 4K, 8K, 16K] Hz.

The incoming signal is firstly gated to reduce noise. The RMS difference of each frequency band between the two sources are then calculated. The results are normalized individually by the bandwidth. The RMS difference of the 10 bands are normalized by removing their mean value. The essential and nonessential bands are determined by the rank of RMS. Higher RMS means that the frequency band is more important. The top 4 bands of the vocal and the top 3 bands of the backing track are considered essential. The K weighting filter [23] is applied before calculating the band energy in order to compensate the uneven human frequency perception. Lastly, the peak filter is applied on the frequency bands that satisfies the three criteria stated above. The gain is the the normalized RMS difference, and the Q is set at 2. The detailed implementation is shown as pseudo code in Algorithm 1. The open-source equalization plugin of choice is MultiEQ[11] by IEM.

4) Reverberation: Moffat and M. Sandler [27] present a set of mixing rules that maps audio features to reverb parameters. Music library analysis [28] shows that music tempo ranges from around 65-200 BPM, with the mean around 120 BPM. The vocal reverb time of The MUSDB18HQ dataset [25] is extracted by the audio plugin Chameleon[12] by Accentize. The reverb time ranges from 0.17 to 4.11 seconds. The proposed equation that linearly maps the tempo to the reverb time,

$$RT = -\frac{1}{45}T + \frac{40}{9} \tag{1}$$

RT is the reverb time, and T is the tempo. De Man et al. [29] show that the reverb loudness are typically -14LU lower than the full mix. In the implementation, the dry/wet ratio is set to 60%. Other parameters are set to constant values. The open-source equalization plugin of choice is FdnRverb[13] by IEM.

B. Data-driven Level Balance

The data-driven approach aims to train a regression model which is able to predict the ideal mixing parameters based on the raw audio features of the input signal. the mixing parameter data for training are gathered from professional mixes. For the task of level balance, the targeted relative loudness as the immediate feature acts as the ground truth. The steps to converted the predicted relative loudness to the gain parameter is the same as the rule-based method described before. Figure 1 is a diagram that shows the processes of deep learning model training and inference.

The training and validation data is the professional produced vocal tracks and backing tracks from MUSDB18HQ dataset [25]. The training set has 100 songs, and the validation set has

[11]IEM MultiEQ, https://plugins.iem.at/docs/plugindescriptions/

[12]Chameleon, https://www.accentize.com/chameleon/

[13]IEM FdnRverb, https://plugins.iem.at/docs/plugindescriptions/

50 songs. The targeted ground truth is the relative integrated loudness [23] between the vocal and the backing track of the entire song measured in LU. The input feature is the mel-spectrogram of the mono vocal and the backing track at the sample rate of 44100Hz. The FFT size is 2048 with the hop size of 1024. The number of mel filterbanks is 128. Each input feature block includes 64 mel-spectrum temporal steps, and each block represents around 1.51 seconds. The size of the input block is (2, 128, 64). Each input block channel is normalized to its maximum value, and a random gain is multiplied to the channel. The neural network architecture is shown below.

Conv2d (input channels = 2, output channels = 8)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 8, output channels = 16)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 16, output channels = 32)
BatchNorm
ReLU
MaxPool2d
Conv2d(input channels = 32, output channels = 64)
BatchNorm
ReLU
MaxPool2d
Dropout (p = 0.3)
MLP (input features = 768, output features = 1)

For all the convolution layers, the kernel size is 3, the stride is 1, and no zero-padding applied. The kernel size of the max pooling layers is 2. The learning rate is 0.0005, and the batch size is 25. The Adam optimizer [30] is used for training. Mean squared error is used as the loss function.

IV. Reuslts and Analysis

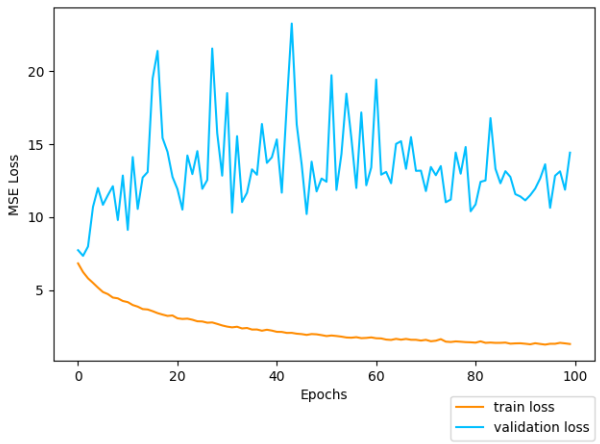

Fig. 2: Train and validation loss

The absolute error is 1.14 dB and 3.80 dB for training and validation after 100 epochs. The figure 2 shows the mean squared error over the training. Although the training loss decreases steadily, the validation loss indicates that the model is not learning at all. The failure of the trial experiment may be due to the following reasons: (1) the lack of generalization ability of this model; (2) not enough training data; (3) improper input features; (4) improper ground truth representation; (5) the challenge of the task itself; (6) code errors.

(1): the lowest validation loss throughout the training is at the 2nd epoch hints that the model may have overfitted early. The solution can be increasing the complexity and the size of the model while employing more regularization techniques. (2): datasets of processional or semi-professional mixes like MedleyDB [31], iKala [32], ccMixter [33], and Rock band [34]. Source separation tools can also be used to extend the dataset from almost unlimited published recordings. (3): The current input feature represents the audio of 1.51 seconds. The segment may be too short, and features other than mel-spectrogram can also be taken into consideration. (4): The current ground truth is the relative loudness between the vocal and the backing track of the full song measured in dB. This representation can be changed into the local relative loudness of the audio segment. The loudness in dB can also be converted to amplitude or scaled down to [0, 1] by min-max normalization. (5): Neither the training set or the validation set represents the best mixing choice since *best* cannot even be defined in a creative task. The failure of the deep learning model is expected, and it is still meaningful result to understand automatic mixing. However, the model output is not yet processed and create a mix. The averaged prediction over the song and the listening test may show a different result. (6): The code errors in training the model and data processing must be eliminated before evaluating the model. A trial run of this model on a separated task may prove the reliability of both the code and the model.

V. Conclusion

A rule-based system for vocal and backing track automatic mixing is implemented. Based on literature and data analysis, the system can perform level balance, compression, equalization, and reverberation. A listening test will be conducted to evaluate the effectiveness of this system. A trial experiment of using convolution neural network to perform level balance shows that the model is not learning. Future work is needed to improving the data-driven level balance model.

References

[1] D. Moffat and M. B. Sandler, "Approaches in intelligent music production," in *Arts*, Multidisciplinary Digital Publishing Institute, vol. 8, 2019.

[2] J. G. Schloss, *Making beats: The art of sample-based hip-hop*. Wesleyan University Press, 2014.

[3] M. Stassen, *Having paid out $150m to creators, BeatStars launches Sony Music Publishing-backed publishing service*, 2021. [Online]. Available: https://www.musicbusinessworldwide.com/having-paid-out-150m-to-creators-beatstars-launches-sony-music-publishing-backed-publishing-service/.

[4] X. Zhou and F. Tarocco, *Karaoke: The global phenomenon*. Reaktion Books, 2013.

[5] J. D. Reiss, "Intelligent systems for mixing multichannel audio," in *2011 17th International Conference on Digital Signal Processing (DSP)*, IEEE, 2011.

[6] E. Perez Gonzalez and J. Reiss, "A real-time semi-autonomous audio panning system for music mixing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–10, 2010.

[7] B. De Man, J. Reiss, and R. Stables, "Ten years of automatic mixing," in *3rd Workshop on Intelligent Music Production*, vol. 15, 2017.

[8] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production*. Audio Engineering Society Presents. Taylor Francis, 2019.

[9] D. Moffat, "Ai music mixing systems," in *Handbook of Artificial Intelligence for Music*, Springer, 2021, pp. 345–375.

[10] E. T. Chourdakis and J. D. Reiss, "Automatic control of a digital reverberation effect using hybrid models," in *Dereverberation and Reverberation of Audio, Music, and Speech Conference*, Audio Engineering Society, 2016.

[11] E. T. Chourdakis and J. D. Reiss, "A machine-learning approach to application of intelligent artificial reverberation," *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, 2017.

[12] D. Barchiesi and J. Reiss, "Reverse engineering of a mix," *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, 2010.

[13] J. T. Colonel and J. Reiss, "Reverse engineering of a recording mix with differentiable digital signal processing," *The Journal of the Acoustical Society of America*, vol. 150, no. 1, 2021.

[14] D. Moffat and M. Sandler, "Machine learning multitrack gain mixing of drums," in *Audio Engineering Society Convention 147*, 2019.

[15] M. A. Martınez Ramırez and J. D. Reiss, "Deep learning and intelligent audio mixing," in *3rd Workshop on Intelligent Music Production*, vol. 15, 2017.

[16] M. Martinez Ramirez, D. Stoller, and D. Moffat, "A deep learning approach to intelligent drum mixing with the wave-u-net," Audio Engineering Society, 2021.

[17] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *Proceedings of ISMIR*, 2018.

[18] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *ICASSP*, IEEE, 2021.

[19] J. Engel, C. Gu, A. Roberts, *et al.*, "DDSP: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2019.

[20] M. A. Martínez Ramírez, O. Wang, P. Smaragdis, and N. J. Bryan, "Differentiable signal processing with black-box audio effects," in *ICASSP*, IEEE, 2021.

[21] A. Case, *Mix Smart: Professional Techniques for the Home Studio*. Focal Press. Taylor Francis, 2011.

[22] B. Man, B. Leonard, R. King, J. D. Reiss, *et al.*, "An analysis and evaluation of audio features for multitrack music mixtures," in *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR 2014)*, 2014.

[23] EBU–Recommendation, "Loudness normalisation and permitted maximum level of audio signals," 2011.

[24] E. T. 3342, "Loudness range: A measure to supplement ebu r 128 loudness normalisation," 2016.

[25] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *MUSDB18-HQ - an uncompressed version of musdb18*, Dec. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373.

[26] S. Hafezi and J. D. Reiss, "Autonomous multitrack equalization based on masking reduction," *Journal of the Audio Engineering Society*, vol. 63, no. 5, 2015.

[27] D. Moffat and M. Sandler, "An automated approach to the application of reverberation," in *Audio Engineering Society Convention*, 2019.

[28] S. Cottrell, "Big music data, musicology, and the study of recorded music: Three case studies," *The Musical Quarterly*, vol. 101, no. 2-3, pp. 216–243, 2018.

[29] B. De Man, K. McNally, and J. D. Reiss, "Perceptual evaluation and analysis of reverberation in multitrack music production," *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, 2017.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015.

[31] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive mir research.," in *ISMIR*, vol. 14, 2014.

[32] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, *et al.*, "Vocal activity informed singing voice separation with the ikala dataset," in *ICASSP*, IEEE, 2015.

[33] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, 2014.

[34] Harmonix, *Rock Band*, MTV Games, Electronic Arts, 2008.